10-2006

Performance Rights for Software

Mark Perry
University of Western Ontario, mperry@uwo.ca

Stephen M. Watt
University of Western Ontario, watt@csd.uwo.ca

PERFORMANCE RIGHTS FOR SOFTWARE

Mark Perry & Stephen M. Watt

Department of Computer Science
University of Western Ontario
London Ontario, Canada N6A 5B7
{markp,watt}@csd.uwo.ca

ABSTRACT

As we use software in increasingly varied contexts, the concept of a software license has become progressively more complex. Software is embedded in devices that do not obviously resemble computers. Web services make software on one computer available to anyone with internet access. An individual may use several computers over the course of the day so the concept of a node locked or individual license is no longer clear. How should time based and single use and consumptive licenses be governed and interact? This paper examines how these and other issues in software licensing can be seen as instances of the general concept of performance rights, rather than simply reproduction rights. Licenses involving finely specified performance rights are common in the entertainment industry for music, film, stage and television. We describe how, as software and our use of it becomes more sophisticated, we see performance rights as becoming an apt basis for software licensing.

1. Introduction

From the moment one person writes a piece of software and a second person wants to use it, the questions of ownership, rights and terms of use arise. In our society, these terms of use are expressed within the framework of intellectual property rights, and these rights may be sold, assigned, waived or licensed.

In the world of software, several sorts of intellectual property rights have been used as the basis of the chain of agreements between the creators of software and the ultimate users. These have included copyrights, patent rights and moral rights. Copyright reproduction rights govern the right to make copies of a work. Patent rights address the right to manufacture and sell goods or services that make use of particular innovations. Moral rights ensure that the creator of a work has the right to be identified as such, and can dictate that it be used in ways that preserve its artistic integrity.

All of these rights have problems if considered as the basis for governing software use. In this paper, we propose that *performance rights* are perhaps a better fit as the umbrella intellectual property right to support the licensing of software. We see strong analogies with other creative works protected by performance rights including film, video, music and stage productions and find analogues between software and these other media. We see an increasing similarity between the creation of large software projects and film productions, hinting that performance rights provide a natural legal framework under which to license software of all kinds.

The outline of this paper is as follows: in Section 2, we discuss the protection of software through copyright reproduction right and problems with this model. Then, in Section 3, we explore recent trends in software licensing that attempt to deal with these defects Section 4 reviews the concept of performance rights and observes how these can be applied to software. Section 5 discusses the relevant international conventions covering performance rights. Finally, in Section 6, we offer several models under which performance rights and software may be licensed. We then give some concluding thoughts on the adoption of performance rights as a model for the software business.

2. Software and the Reproduction Right

Initially the legal concepts of ownership and right of use were tied to physical objects, places or things. For example, in many societies one can own a pair of shoes or have hunting rights on a particular piece of land. As our society has evolved, many of our agreements have come to deal with increasingly abstract rights, such as the right to buy shares in a certain company at a certain price at a specific point in the future or the exclusive right to broadcast electro-magnetic transmissions at a certain frequency. The right to restrict the making of copies of a work is an abstract paradigm.

In some cases, there is a high intrinsic value in an object and also thus in its copy. For example, the value is clearly exhibited with a quantity of a disease-curing vaccine, or seeds of a new variety of crop plant. In other cases, the physical material of the object being copied, the paper and ink constituting a novel or the plastic of a DVD, is largely irrelevant. Instead, it is the information content that is of worth, be it the sequence of words on a page, the layout of pigments on a canvas or the organization of numbers in a computer program. Though of social value, there is little economic value in information where reproduction and transaction costs approach zero, but there are many reasons

to give information such value by protecting it and creating an artificial scarcity.[i] The utilitarian perspective, for example, justifies copyright protection to stimulate the creativity and effort that goes into making the original expression.

Before the invention of the printing press, the act of copying a written work was not only labour intensive, but also the raw materials required for the copy held significant value. Now, for many works, the act of copying is almost effortless, and the transaction cost minimal.[ii] Copyright has therefore naturally become more important. This is particularly true for digital media: a digitized song can be transferred from a compact disc to a computer's hard disc or copied over a network with no direct human involvement.

The present ease of making copies is necessitated by the way the works are *used*. For example, simply looking at a CD does not reflect its true value--one has to play it. Playing it involves reading the information off the disc using a CD player. In the CD player, there is never a complete copy of the song as it is played, as the 'track' is accessed dynamically. When a song is 'ripped' to a computer, an entire copy is transferred to the computer, and there is another reproduction of the song. It is in this type of situation where the reproduction right begins to break down for software. While we can argue that one need not make an entire copy of the song in order to play it, modern operating systems must make copies of programs in order to execute them.

Data or programs used by computers are subject to many acts of copying not explicitly contemplated in copyright agreements. For example, the copying of a computer program from the hard drive into memory in order for it to be executed, or copies made in scheduled backups of a computer system.[iii] Additionally, it is common for sections of programs to be copied to a paging disc during program execution. In a modern multi-processing environment, there may be multiple instances of an entire program on the paging disc at any given time. The problems of applying copyright law to software are well documented, with solutions varying from the introduction of *sui generis* to relying on the patent system for its protection from free-riding.[iv] However, these solutions do not offer continuity of protection nor do they offer pertinent protection measures, such as under performance. The current *status quo* of software protection under copyright reproduction rights is dictated largely by the provisions of TRIPS, [v] and there is increasing demand for specialized protection of technology protection measures (which is usually as software) under WCT and WPPT agreements. [vi]

We see that restrictions on copying are very heavily dependent on the technology of the day. To assure that a program runs well and makes only permitted copies on a particular computer architecture might require new kinds of copies be made on a new computer architecture. These kinds of copies can be anticipated by neither the legislation nor the licensor at the time of the license grant. A broad license permitting new kinds of copies would almost certainly allow unintended and unfair use, while a narrow license forbidding copying not explicitly allowed would make fair use of software on newer computer models a license violation.

Typically, proprietary license agreements require all copies be destroyed on termination of the agreement (or even on the change of hardware in some circumstances) but these restrictions are practically unenforceable since many large corporations have staged backups that end with large collections of material on read-only volumes for permanent retention. Licensed material, including software, is included in such backups, of which there may be many copies per month. It would be impossible, or perhaps merely expensive and unsafe, to attempt to modify a permanent backup volume to delete some of the information to comply with anti-reproduction rights.[vii]

3. Licensing Trends

Even if only one traditional copy of software or data is made, the creator may wish to place restrictions on who may use it, and how. Software licenses are therefore evolving to include complex terms of use.

Many licenses are restrictive: users are only allowed to use on a particular machine or the number of simultaneous licensed users is limited.[viii] Both the issues of copying for execution and who may use are particularly important for works placed on a networked computer. The development and adoption of the service and utility computing models can only exacerbate these existing licensing problems. From the software licensing perspective, a program that may be executed on a server from a remote location raises the problem of unrestricted use rather than the number of ephemeral copies involved in that use.

Software licenses from the 1980s typically did not contemplate and therefore in principle may be placed on an internet server and used by everyone in the world. Modern licenses usually have explicit clauses to limit this use, or at least derive more revenue from it. Another example of a modern licensing technique is to limit the number of times a program may be used. For example, in the Canadian version of the QuickTax program (from Intuit Canada) users are limited to the preparation of six tax returns on incomes over $25,000. This is true even when the software is installed on only one computer and used by only one person.

We see that while modern software licenses are typically built around the reproduction right in copyright as a 'literary work', this is a rather artificial protection mandated by the TRIPS Agreement and implemented in the Copyright Act. The central issue is actually the conditions under which the software may be used. That is, who may use it, under what conditions and when. We claim that running a computer program is very similar in principle to playing a piece of recorded music or video, or indeed a radio broad-

cast for web services. While the particular recording of a piece of music, motion picture or stage play script is protected by one copyright or another, in principle the more important intellectual property right in these works in this century is the exclusive right to authorize the performance of the work. [ix] This is particularly evident when the primary revenue stream is seen to come from performance licensing. We argue that these rights should equally well apply to software.

4. Performance Rights

An area of intellectual property that has been developing rapidly over the last few years is that of protection for performance. The "Performance rights" in a work capture the notion of when and how a work may be performed. For example, whether a play may be performed in front of a live audience or a piece of music may be broadcast by radio or played as the "on hold" music for a telephone answering desk.

Contracts that license performance rights can very finely specify the particulars of when, how and under what conditions the work may be performed. For example, the timing of opening nights of major motion pictures is carefully coordinated contractually. When one purchases a DVD, one does not have the right to use this arbitrarily. Consumer DVDs are licensed for private home viewing only, even if one owns the DVD and one does not make any copy whatsoever. Contracts for stage production of Mamet's play Glengarry Glen Ross [x] exclude its performance as theatre in the round.

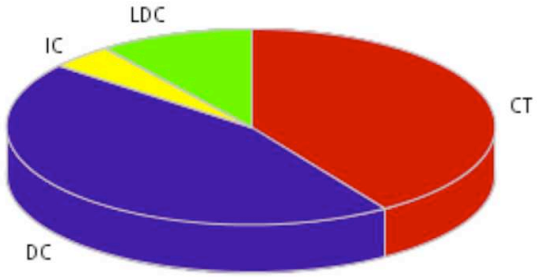
The few legal actions that have made the courts and have turned on the performance right have mostly been driven by collective societies, with an example being *Society of Composers, Authors and Music Publishers of Canada (SOCAN) v. Demers* [xi] a case that is illustrative of a common situation. The defendants had authorised the performance of musical works in their bar even though they did not have a licence for those performances. Demers had to pay royalties, plus profits made from the infringement, and exemplary damages: they had refused to obtain appropriate licences even after being warned by SOCAN [xii] that their actions were infringing the Copyright Act.

Despite the paucity of legal action, the performance right is an important one. The National Music Publishers' Association most recent survey reports that some 46% of the music industry's income was from performance royalties, outstripping the 40% from reproduction based income.[xiii] With the reduction in the need for physical media, this disparity is likely to grow.

The performance right has a shorter history than other intellectual property rights. Traditionally performance was protected as *droit voisins* or one of the 'neighbouring' rights akin to copyright. The International Convention for the Protection of Performers, Producers of Phonograms and Broadcasting Organizations, known as the Rome Convention, secured the protection of performers rights (amongst others) . [xiv] In Canada the 1996 amendments [xv] to the Copyright Act extended for broadcasts and performance. The Copyright Act defines performance: [xvi]

> "performance" means any acoustic or visual representation of a work, performer's performance, sound recording or communication signal, including a representation made by means of any mechanical instrument, radio receiving set or television receiving set;

It is important to note that each performance is protected, and that performance rights can, and usually do, co-exist with other rights in a work (for example the protection of an underlying literary work and musical work). Recent developments that illustrate the growing importance of these rights led to the World Intellectual Property Organisation (WIPO) Performances and Phonograms Treaty (WPPT) of 1996 [xvii].



CT: Countries in Transition 41%:
DC: Developing Countries 45%
IC: Industrialized Countries 4%;
LDC: Least Developed Countries 10%

Figure 1 Contracting Parties by Development Status (Countries only) [xviii]

It may seem anomalous that few industrialized countries have as yet adopted the WPPT, however the extension of performers' rights in that treaty (such as moral rights akin to those in literary works) and the greater investment by the industrialized nations have in the established framework for intellectual property protection, regardless of its degree of effectiveness or flaws, goes some way to explain this. The Canadian government is currently resurrecting the Bill for introduction to the House of Commons that will adopt the provisions of both the WPPT and the WIPO Copyright Treaty [xix]. Other WIPO treaties are under negotiation that potentially impact software. The Draft Treaty for the Protection of Broadcasting Organisations, that will give a copyright like right to broadcasters in materials that they transmit [xx].

5. Performance Rights: Models for Software

Clearly there is sufficient flexibility in the concept of performance rights to encompass the needs of software licens-

ing. To further illustrate how performance rights in software are a natural framework to govern software's use, consider the following hypothetical family of related works: a stage play *The Dukes of Wrath*, the cinematic production, the Broadway musical, the soundtrack of the Broadway musical, the anime version, the Macromedia Flash cartoon and the Java program rendering the same cartoon. In each case, there is the abstract work itself, the physical copy of the work and the performance of the work. Each of these is different and each of these is governed by different legal theories. To further underscore that computer programs are works that are performed, we point to the programming language Shakespeare. See Appendix A for a play that is actually a machine executable Shakespeare [xxi] program to compute the Fibonacci numbers.

Traditional licensing models for works intended for entertainment tend to focus on public performance. Clearly, in their case, significant potential revenue comes from paid admission to public events or businesses using the materials and exposing their clientele, as public, to these works. One can also certainly imagine performance rights in music based on a pay-per-use private listening model.

Much of the use of software is private. Perhaps the normal software license best corresponds to the private home viewing license on VHS tapes, DVDs and music CDs. If the end user license agreement clearly specifies that all performance rights are reserved, save those that are explicitly granted license, then the next-generation of problems akin to network server use will be avoided. The end user has the right to perform through mechanical or other means the work only in accordance with the license agreement. Only the licensed user may perform the work.

We list below a few example terms under which performance rights may be of how performance rights in software could be specified:

- The work may only be performed for the licensee.
- The work may be performed for any employee of the licensee.
- The work may be performed on only a specified computer.
- The work may only be performed once, or a certain number of times, or an unlimited number of times during a particular period.
- Performance from a server over the Internet.
- Pay-per-use performance from a server over the Internet.
- Pay-per-period subscription of performance for a period over a network.
- Performance only for users of legal age (For example, gambling, sexually explicit software).

Licensing through agents or collective licensing analogous to SOCAN, BMI or ASCAP is a useful model to adopt, particularly with utility model computing services. This formulation solves the expensive problem of the eradication of backup copies since the licensee will not be required to destroy all copies at the end of the license but only to warrant never to perform them. Other restrictions on performance could relate to maintaining the artistic integrity of the work. For example, that a piece of software not be performed at a computer less than 3GHz is completely analogous to staging restrictions in the performance rights of a play. Naturally, software vendors are not typically interested in restricting the use of their software, only the unpaid use of their code. Thus, the main thrust from a business perspective, is to enable a model that better reflects the charging for use – ie performance of the code.

If a party is found to have violated copyright in a work then the remedies that may be sought are based on damages or accounting for the profits that flow out of the creation and existence of the infringing components of the work. Destroying the copies does not effectively remedy the damages to the copyright holder. A violation of terms for performance rights, however, has a clearer basis for remedy as quantization of the value of a performance would be easier. Looking forward, we see that computer software is intended to be used. Unexecuted, a copy of software may perhaps have literary worth, instructional worth or artistic worth, but it is through performance that software provides its true value. By protecting the performance of the code in addition to the value of the work from reproduction as a literary work, the intellectual property framework will better reflect the nature of computer software. With the adoption of utility models, software service centres online, and other remote execution of code, the performance model will provide a more realistic and true reflection of the use of this resource.

6. Conclusions

We have argued that the current model for intellectual property rights in software is inadequate: although the copyright reproduction right and patent rights address the manufacture of software they do not capture the concepts necessary for software licensing – particularly with the trend of making more software available as web services and centralized servers. We contend that the awkward terms of use in current software licensing practices are built on shaky legal grounds with little theory to withstand challenges. In contrast, the licensing of performance rights is well settled and suited to a remote distribution model as well as on fixed media (whether radio or compact disk). The use of computer software amounts to a performance of a work and thus falls squarely within this framework. With the adoption of utility models, software service centres online, and other remote execution of code, the performance model will provide a more realistic and true reflection of the use of this resource. The model presented here supports licenses derived from the true value of the software rather than an artificial distinction based on incidental copying. By basing software licenses on a well established legal theory with precedent that are supported by international conventions, we anticipate that licensing software through performance rights will be more robust and be

better be able to withstand challenges compared with those based on ad hoc terms.

Appendix A: *A Shakespeare Program*

The following play is a program written in the Shakespeare programming language, version 1.2.1. It reads an integer *n* as input and prints out the first *n* Fibonacci numbers.

```
                The Rabbits of Verona.

Proteus, a gentleman of Verona.

Valentine, the best friend of Proteus.

Queen Elinor, who decides where it will end.

Isabella, a Countess who starts from nothing
and becomes equal to the Queen.

The Abbot of Westminster, who has a short
memory.


    Act I:   The origin of the specious.

   Scene I: Gentlemen of courage and envy.

[Enter Proteus and Valentine]

Valentine: You are a hero.

Proteus: You are nothing.

[Exeunt]


     Act II:  Counting on the Countess.

        Scene I: A secret encounter.

[Enter Isabella and Queen Elinor]

Isabella: Listen to your heart!

Queen Elinor: You are nothing.

      Scene II: A secret conversation.

Queen Elinor: Am I better than you?

Isabella: If not, let us proceed to scene V.

[Exeunt]

    Scene III: Gentlemen exchange views.

[Enter The Abbot of Westminster and Proteus]

Proteus: You are as bold as myself.

[Exit The Abbot of Westminster]

[Enter Valentine]

Valentine: You are as brave as the sum of
myself and yourself.

[Exit Proteus]

[Enter The Abbot of Westminster]

The Abbot of Westminster: You are as smooth
as me. Open your heart!

Valentine:  You  are  as  misused  as  an
honest old rural bottomless rich grandfather.
Speak your mind!

[Exeunt]

    Scene IV: The Countess gains stature.

[Enter Isabella and Queen Elinor]

Queen Elinor: Thou are as bold as the sum of
thyself and a roman.

Isabella: Let us return to scene II.

[Exeunt]

     Scene V: The Countess is now great.

Queen Elinor: You are as mighty as an angel.

Isabella: You are as miserable as a toad.

[Exeunt]


         Act III: Untying the knot.

   Scene I: The confession of a gentleman.

[Enter Proteus and Isabella]

Isabella: Open your heart!

Proteus: You are as lovely as the sum of the
sweetest reddest blossoming rose and a sunny
summer's day.  Speak your mind!

[Exeunt]
```

After compilation, the program may be run to compute the Fibonacci numbers as shown in the following log. The number 45 is typed by the user to specify how many Fibonacci numbers are desired.

```
% rabbits

45

1 1 2 3 5 8 13 21 34 55 89 144 233 377
610 987 1597 2584 4181 6765 10946 17711
28657 46368 75025 121393 196418 317811
514229 832040 1346269 2178309 3524578
5702887 9227465 14930352 24157817
39088169 63245986 102334155 165580141
267914296 433494437 701408733
1134903170 1836311903
```

References

[i] The Canadian Journal of Law and Jurisprudence, Vol. XVI, Number 2 (July 2003), with many papers exploring the foundations for intellectual property protection.

[ii] The World Intellectual Property Organisation (WIPO) Copyright Treaty has been followed by a flurry of digital centric legislation, such as that under consideration by the Canadian government .

[iii] Indeed, this is recognized in the s.30.6 Copyright Act (C-42) that permits the reproduction of a computer program for the purposes of operability.

[iv] Pamela Samuelson "Contu Revisited: The Case Against Copyright Protection for Computer Programs in Machine--Readable Form", 1984 Duke L.J. 663 (early discussion calling for sui generis protection of object code); Computer Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 712 (2d Cir. 1992) (squeezing software into the copyright mold); Jane C. Ginsburg, Four Reasons and a Paradox: The Manifest Superiority of Copyright Over Sui Generis Protection of Computer Software, 94 Colum. L. Rev. 2559 (1994); Michael A. Dryja "Looking to the Changing Nature of Software for Clues to its Protection", 3 U. Balt. Intell. Prop. L.J. 109; and James P. Chandler, "Patent Protection of Computer Programs" 1 Minn. Intell. Prop. Rev. 2.

[v] The Trade-Related Aspects of Intellectual Property RightsAgreement is Annex 1C of the Marrakesh Agreement Establishing the World Trade Organization, signed in Marrakesh, Morocco on 15 April 1994. Article 10 provides that computer programs be protected in the same way as literary works are under the Berne Convention (1971).

vi Full text of WIPO Treaties can be found at http://www.wipo.int . As of 12 May 2006, 57 nations had signed up to the WPPT.

[vii] Although in some jurisdictions, such as under the s.50 United Kingdom Copyright Designs and Patent Act 1988, as amended, it is permissible to make 'necessary or lawful' backups, in most jurisdictions the ability to legally backup outside a license is restricted.

[viii] There are many software licenses, but these can be seen as 17 major types, as identified in H Kaminski & M Perry "Pattern Language for Software Licensing" In Proceedings of 10th European Conference on Pattern Languages of Programs (Euro Plop 2005), 2006.

[ix] For example, in 2005 the American Society of Composers, Authors and Publishers (ASCAP), reported a revenue increase of 6% in 2005 (total of $749 million), in royalty payments to members topping $645 million. ASCAP has the largest portfolio in the US, and collects for members for performance of their repertory www.ascap.com/press/2006/031306_financial.html .

[x] D. Mamet. Glengarry *Glen Ross*. 1984. Grove Press (reissue edition) 1992.

[xi] [1998] F.C.J. No. 1076

[xii] The Society is a "Canadian copyright collective that administers the performing rights of more than 75,000 composer, author and publisher members, and the hundreds of thousands of members of our affiliated societies worldwide, by licensing the use of their music in Canada." One of their activities is to "collect licence fees on [members'] behalf and distribute royalties to them." http://socan.ca .

[xiii] The National Music Publishers' Association, Inc. & The Harry Fox Agency, Inc "International Survey of Music Publishing Revenues" (2001) http://www.nmpa.org . Similar trends can be seen from more recent United Kingdom revenue figures. The Performing Rights Society at www.prs.co.uk reports that in 2004 they distributed 256.2 M pounds sterling to composers, songwriters and music publishers. See also the ASCAP http://www.ascap.com/ .

[xiv] The Rome Convention for the Protection of Performers, Producers of Phonograms and Broadcasting Organizations, is administered by WIPO: http://www.wipo.int/treaties .

[xv] Introduced by Bill C-32 1996 (Canada).

[xvi] Section 2 Copyright Act 1985 (Canada).

[xvii]Above, note [vi].

[xviii] Reproduced from http://www.wipo.int .

[xix] The Bill introduced in 2005 died with the fall of the government. However, the new government seems to be intent on introducing similar legislation. The proposed direction for reform (unchanged as yet) is at http://pch.gc.ca/progs/ac-ca/progs/ pdacpb/reform/statement_e.cfm

[xx] Above, note [vi].

[xxi] The Shakespeare Programming Language was designed by Kalle Hasselström and Jon Åslund with the stated goal of making program source code resemble Shakespeare plays. Documentation and an implementation for the programming langauge may be found at http://shakespearelang.sourceforge.net